	Disclosure ARC8-1999-0120
	Created By: Mike Emens Created On: 04/14/99 11:31:49 AM
	Last Modified By: Mike Emens Last Modified On: 04/28/99 03:17:07 PM
	*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Summary

Status	Under Evaluation
Processing Location	ARC
Functional Area	DPB - Computer Science - (A.K. Chandra)
Attorney/Patent Professional	Ray Strimaitis/Almaden/IBM
IDT Team	Ray Strimaitis/Almaden/IBM; Cheryl Ruby/Almaden/IBM
Submitted Date	04/28/99 01:13:18 PM
Owning Division	RES
PVT Score	48

RECEIVED

MAY 08 2003

Technology Center 2100

Inventors with Lotus Notes ID's 45 L2XA
49949801 843388 K57E
Inventors: Mike Emens/Cupertino/Contr/IBM, Reiner Kraft/Almaden/IBM

Inventors without Lotus Notes ID's

IDT Selection

IDT Team:	Attorney/Patent Professional:
Ray Strimaitis/Almaden/IBM	Ray Strimaitis/Almaden/IBM
Cheryl Ruby/Almaden/IBM	

Response Due to IP&L : 05/28/99

Main Idea

*Title of disclosure (in English) System and Technique of associating search results with other search results selected in previous search sessions

*Idea of disclosure

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

The invention is related in the area of Internet Search engine technology.

Content of the Internet is increasing and thus it's more and more important to improve current search methodologies. This idea is to **monitor user selections relative to the search string**. With this information we can empower future users in several ways. For instance, people who are searching for similar resources (therefore using the same query string), can leverage from the knowledge of other users. Our mechanism will record user selections along with the query string.

The following example will make the idea more clearer. User A is looking for a particular and efficient Quicksort algorithm. He/She uses the jCentral search engine's "Advanced Powersearch" search form to construct a complex query. The result page will contain 100 external resources

(URLs), which contains hyperlinks to various implementations of the Quicksort algorithm. User A now will begin to read through the abstracts provided and eventually will choose one result item for closer examination. Thus User A selects a hyperlink pointing to the external resource. Typically the document will be downloaded into a viewing device (e.g. a web browser) and then User A is able to further examine the whole document. When User A is done with reviewing the document, he/she might also select other links to resources on the result pages for further review, which look promising. **Our system will record all the hyperlinks selections by that user.** Everytime the user clicks on a hyperlink on a result page, our system will **associate this particular resource with User A's original search query** and store this information (<user query, URL> pair) in a database system.

User B later uses the system independently to also search for a Quicksort algorithm and enters the same query using jCentral "Advanced Powersearch" electronic form. Our system will forward the request to the search engine, which will retrieve the matching resources. However, before returning the matching resources to User B, our system will cross reference each results with other selections made by other users who formed the same search query string. If that result was used by another user performing the same query then a box will appears (called "**See what others found**") to that result for User B. If User B selects that box then the system will return a new list composed of all the other URLs prior users selected when using that same query string. This includes User A selections.

The invention will provide the following benefits:

1) Reading through the abstracts of a result page is a **time consuming** task. Our recording mechanism will organize resources for a particular query. Because users can view and select from prior user selections, they **save time** reading abstracts.

2) The resources are usually sorted by relevance (matching the original query string). Indexing is done mostly **automatically**. In our system we use the **human's ability to evaluate resources** and store this information for further reuse. Users choose to access result items (by clicking on a hyperlink usually), after they evaluated the abstract of a result item and think, that this could be a good match (for the query they issued before). This human knowledge will be automatically collected and can then be reused by other users. Therefore resources, which related are automatically associated.. Thus the **search quality will be improved** by integrate the human's evaluation capabilities.

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

This invention will record user selections for a given query string. subsequent users performing the same query string will be told there are these alternate results. If desired a list of these alternate results are presented. The user can select from thsi alternate result list.



a-power-catsanddogs.bm|

[Example: Power Search by User "A" - Query = "CATS AND DOGS"]



a-power-catsanddogs-results.br

[Example: Search results for User "A" - Query = "CATS AND DOGS"]



b-power-catsanddogs.bm|

[Example: Power Search by User "B" - Query = "CATS AND DOGS"]



b-power-catsanddogs-results.br|

[Example: Search results for User "B" - Query = "CATS AND DOGS"]

The system consists of the following components:

- 1) Session Manager
- 2) Query Manager
- 3) Alternate Results Manger
- 4) Query Database
- 5) Associator

The following provides an system overview and the interaction between the components.



ArchDiag.doc

[Architecture Diagram]

This invention is best described as a **three step process**.

The **Session Manager** component is responsible for intercepting all user input for the purpose of handling each of these steps.

Step #1:

In the initial state of the this invention there are no **Query Database** entries. A user of this system will first submit a query string to this invention. The "Session Manager" will first establish a session ID for this user if none has already been established. It uses standard web technologies to store state and session information (e.g. Cookies, Active Server Pages, etc.). This session ID will be used to match subsequent queries to the result selections made by this user. The query string is then submitted to the search engine to retrieve search results. Steps #2 and #3 below are then perform. At this point in the process those steps are not significant. The net result of those steps are presented to the user for selection. The key this step is as follows. The user will select zero or more of the results from this initial query. Each selection made by the user is first received by the **Session Manager** then routed to the Associator. The **Associator** will receive the session ID, query string and selection made for this user. The **Associator** will first find the result entry in the "Query Database". If none exist is will create one. If will then find the Alternate Results Vector (ARV) for that query string. If none exist it will create one. This completes this step.

At the end of this step this system has effectively recorded all selections made by this user for that

query string. Subsequent queries by any other user using that same query string can now be presented as Alternate Results and selected by that user.

Step #2:

Past the initial step (step #1) of the this invention there are now entries in the **Query Database** , A user of this system will submit a query string to this invention. The **Session Manager** will first establish a session ID for this user if none has already been established. This session ID will be used to match subsequent queries to the result selections made by this user. The query string is then submitted to the search engine to retrieve search results. When the search engine systems returns the search results, it will add the session ID and Query string information to it and forward this to the **Alternate Results**.

The **Alternate Results Manager** will receive the search results, along with the user query string and session id from the **Session Manager**. Each search result is paired with the query string and submitted to **the Query Manager**. The **Query Manager** will first find the result entry in the **Query Database**. If none exists it will return a FALSE status indicating there are no **Alternate Results**. If a results entry does exist a second check is then made for the an Alternate Results Vector (ARV) based on the query string. If none is found it will return a FALSE status indicating there are no Alternate Results. If an ARV entry does exist for that query string it will return a TRUE status indicating there ARE Alternate Results.

Finally, the **Alternate Results Manger** will format the output for the user. Each formatted result with an Alternate Result will have a GUI element (most likely a button) indicating there are alternate results available.

Step #3:

The user has not got the results of her/his query. For each results having an alternate result set there will be a button indicating there are alternate results. The user can now either select a result (as before) or select the Alternate Results button.

If a results is selected either the browser directly goes to that URL or that request is sent to the **Session Manager**. Either way the user is presented with more information on that results as most search engines do today.

If the user selects the "Alternate Results" button that message is sent to the **Session Manager**. The **Session Manager** will forward this session ID, query string and alternate query message to the **Alternate Results Manager**. The **Alternate Results Manager** will now reference the results from the **Query Database** using the query string. All results in that ARV are now retrieved from the search engine directly. All ARV results found in the search engine are then formatted for output back the user.

At the end of this step the user has now been presented with Alternate Results which match to his/her initial query string.

Description of the Alternate Results Vector (ARV):

In its simplest case, the alternate results vector is a single element vector composed of a query string and associated URL list. Each URL keeps a list of Query strings used to get to that URL. Each of those query string entries keeps a list of other URL's also in that pool.

Using this technique, the system can handle any number quantifiers for resource relevance or irrelevance.



FlowChart-Step-1.doc
[Flow Chart - Step #1]



FlowChart-Step-2.doc
[Flow Chart - Step #2]



FlowChart-Step-3.doc
[Flow Chart - Step #3]



slides.doc
[Slides]

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

none that we are aware of

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.
The invention will be implemented in the Grandcentral Station site of portals (xCentral, jCentral). Currently we are working on a prototype, which will be ready for testing probably end of May 1999. For additional information please reference the "SearchScope" white paper by Reiner Kraft, Peter Yim and Michael Emens. The whitepaper shows how this invention will be integrated in a set of techniques to improve search accuracy.

***Critical Questions (Questions 1 - 7 must be answered)**

***Question 1**

On what date was the invention workable? 04/25/99 Please format the date as MM/DD/YYYY
(Workable means i.e. when you know that your design will solve the problem)

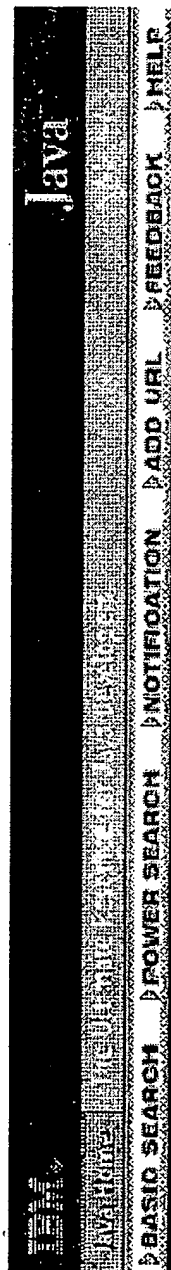
***Question 2**

Is there any planned or actual publication or disclosure of your invention to anyone outside IBM?

☐ Yes
☐ No

System and Technique of associating search results with other search results selected in previous search sessions

User "A" enters "CATS AND DOGS" for search string



Applets

Choose an attribute

All

CATS AND DOGS

Enter keyword(s)

a-power-catsanddogs.bmp

Assume User "A" selects items #1 and #3

(1) <http://www.catsanddogs.com>

Abstract: Why don't cat's and dog like each other? Find out why.....

(2) <http://www.cats.com>

Abstract: Cats and their nine lives. Are there more than nine? Find out...

(3) <http://www.catsanddoggrooming.com>

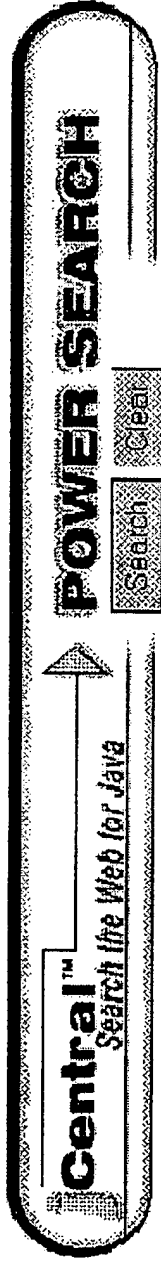
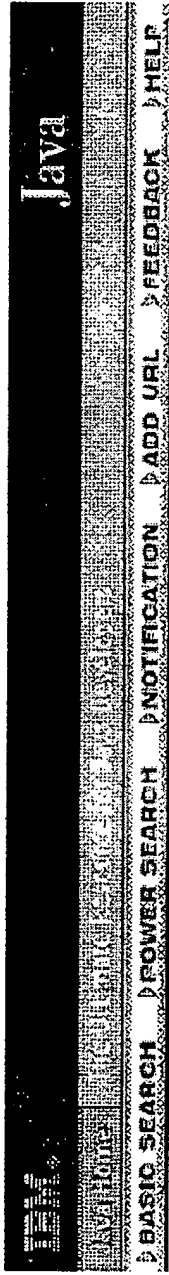
Abstract: Why dogs like water and cats don't.....

(4) <http://www.mansbestfriend-and-ninelives.com>

Abstract: Is dog mans best friend? Or is the cat? Find out who really is....

System and Technique of associating search results with other search results selected in previous search sessions

User "B" also enters "CATS AND DOGS" for search string



Applets

Choose an attribute ▾

All

Enter keyword(s) ▾

CATS AND DOGS

User "B" enters "CATS AND DOGS" for search string

(1) <http://www.catsanddogs.com>

Reuse Similar Results

Abstract: Why don't cat's and dog like each other? Find out why....

(2) <http://www.cats.com>

Abstract: Cats and their nine lives. Are there more than nine? Find out...

(3) <http://www.catanddoggrooming.com>

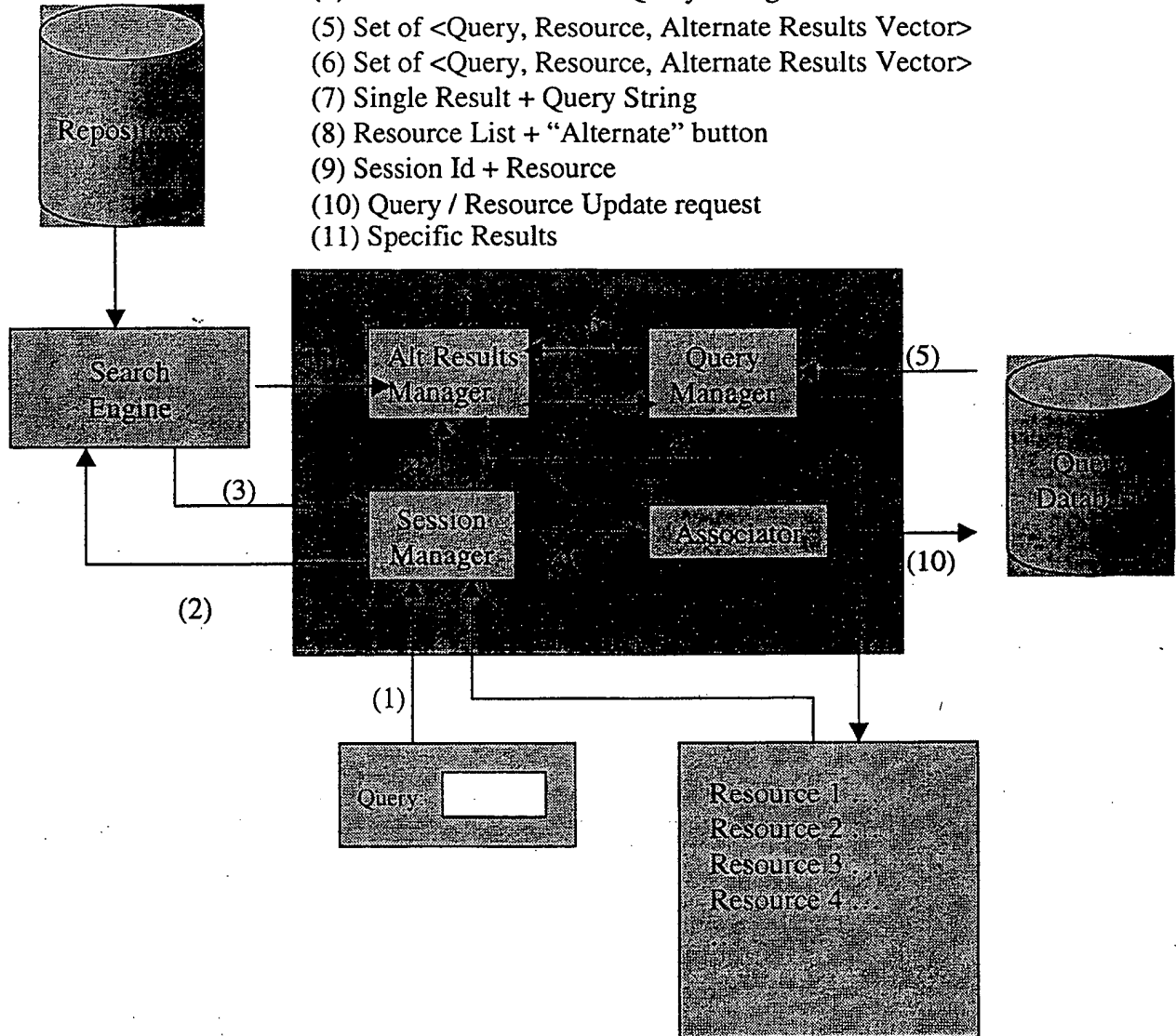
Reuse Similar Results

Abstract: Why dogs like water and cats don't....

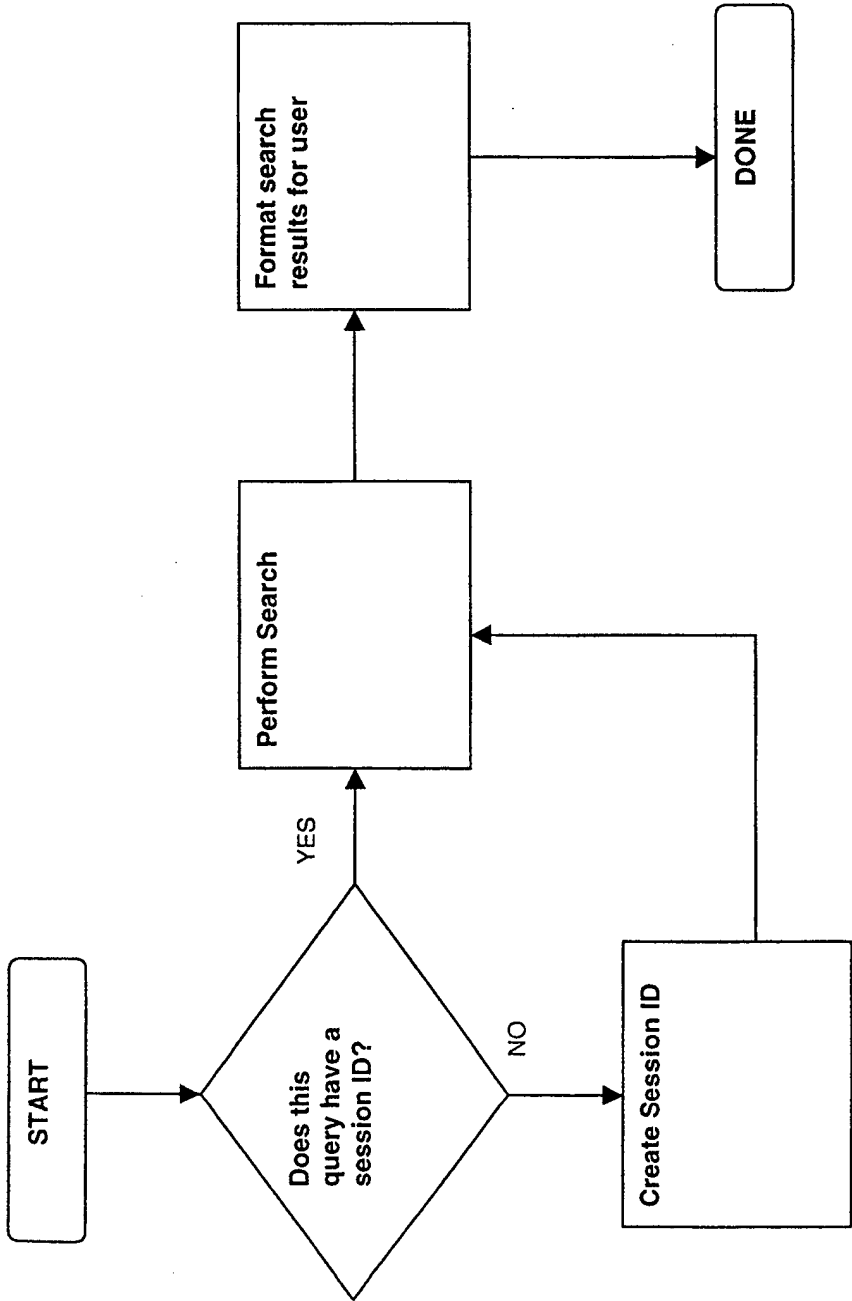
(4) <http://www.mansbestfriend-and-ninelives.com>

Abstract: Is dog mans best friend? Or is the cat? Find out who really is....

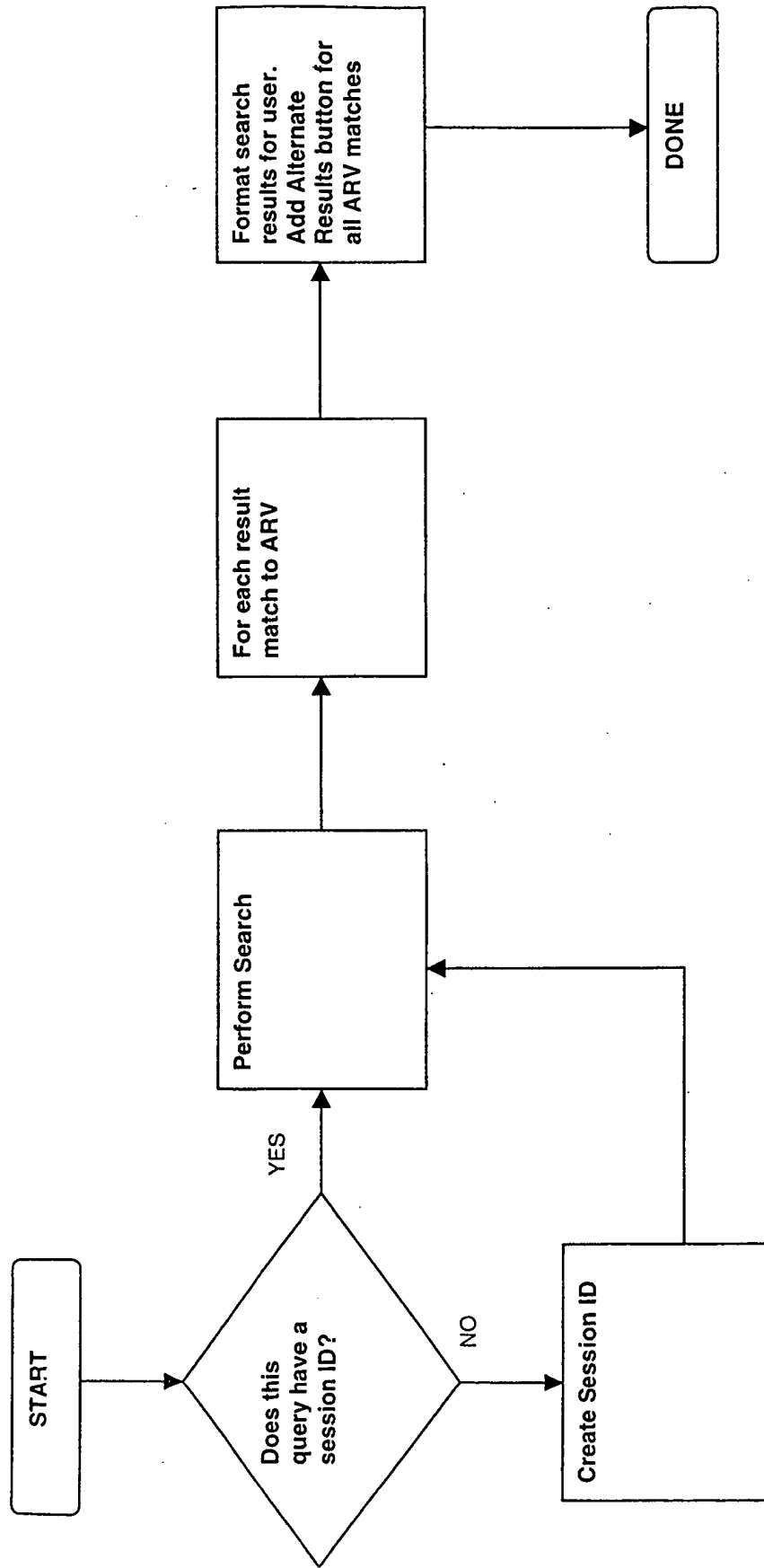
- (1) Query String
- (2) Query String
- (3) Results
- (4) Results + Session Id + Query String
- (5) Set of <Query, Resource, Alternate Results Vector>
- (6) Set of <Query, Resource, Alternate Results Vector>
- (7) Single Result + Query String
- (8) Resource List + "Alternate" button
- (9) Session Id + Resource
- (10) Query / Resource Update request
- (11) Specific Results



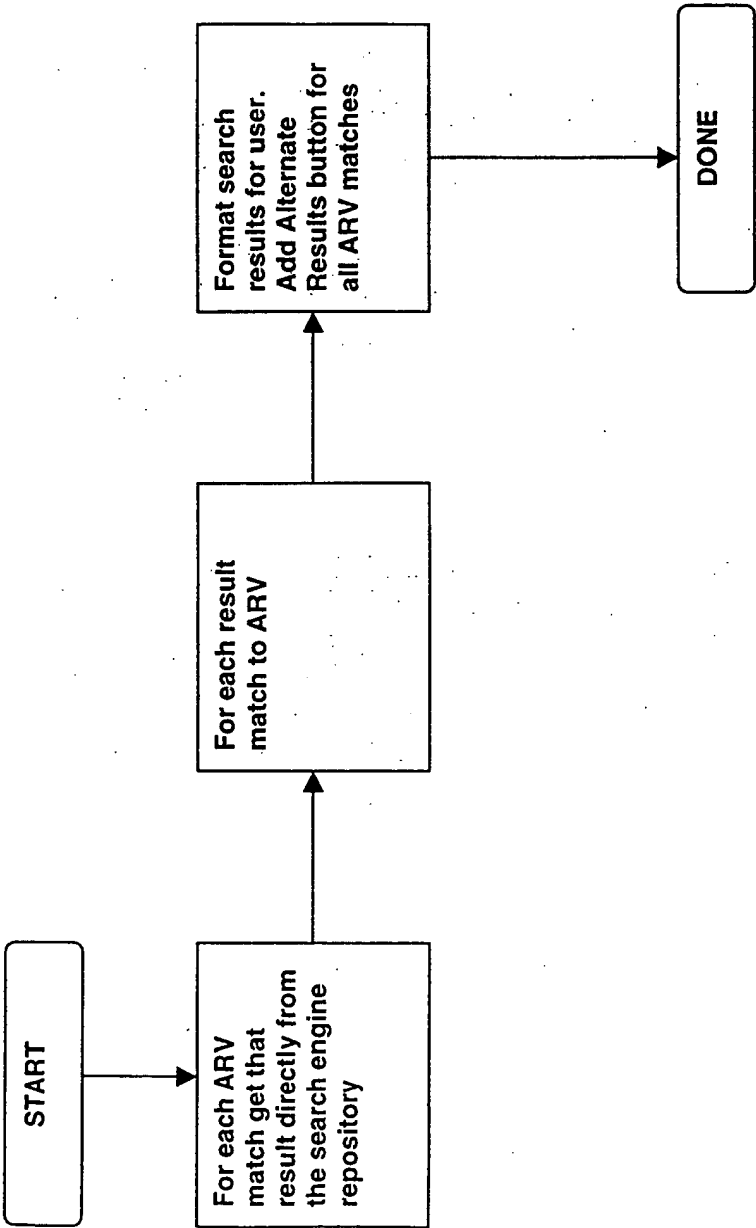
Flow Chart – Step #1



Flow Chart – Step #2



Flow Chart – Step #3



Today's Technology

- ◆ User submits query
- ◆ Search Engine presents results
- ◆ Users select zero or more of those results

The Problem

- ◆ Though most users select one or more results for a given search string. All users have no way of knowing what prior users selected when entering the same search string

This Invention

- ◆ Will record users selections based on that search string.
- ◆ Subsequent users performing the same search can now see that someone else not only selected that result but other results as well.
- ◆ That user can now select from that alternate result list and see what other users selected at the same time of selecting that result.